Global optimization software library for research and education

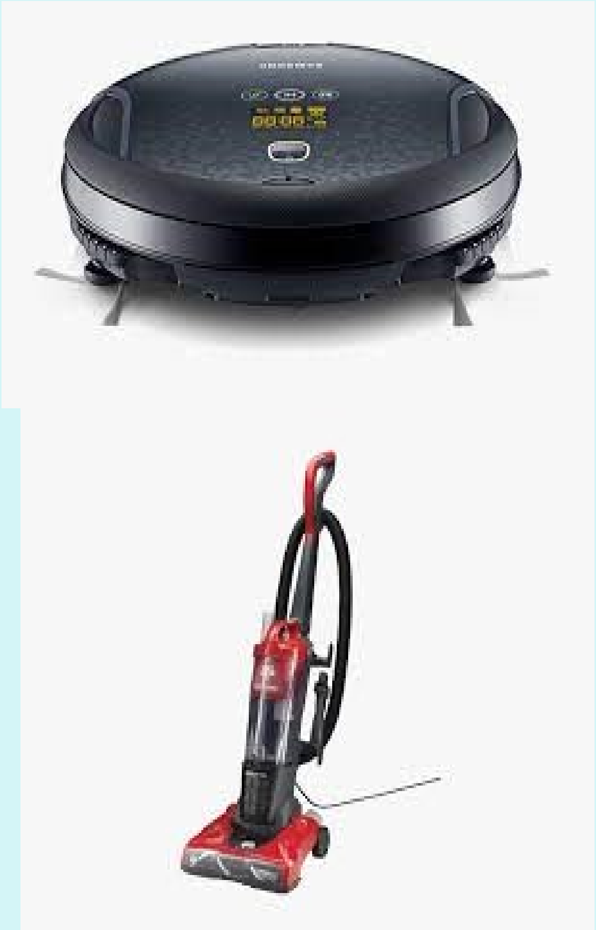
Nadia Udler

- Created for students and researchers
- Has its roots in potential theory
- Optimization methods with desired properties are created based on basic modules, by varying parameters of generalized algorithm
- Allows
  - to design optimization methods with guaranteed convergence but without the use of derivatives of objective function
  - to compare existing optimization methods
  - to understand principles of learning algorithms
  - to design custom variations or hybrids of known heuristic optimization methods.

Selecting cleaning strategy

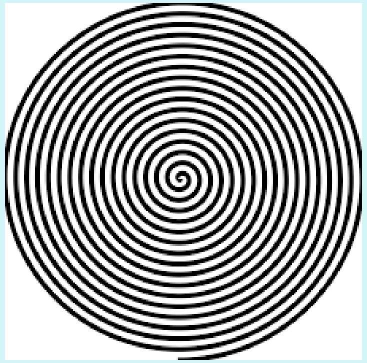
Example 1 : Vacuum cleaners

Goal: cover the entire surface

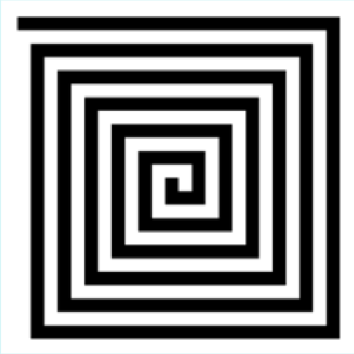


Possible strategies

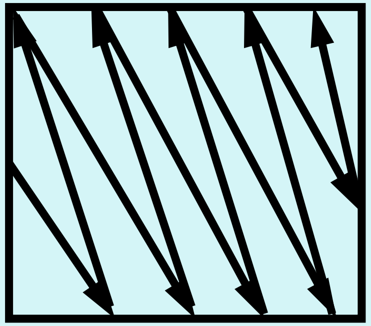
Spiral-like motion from inside



Spiral-like motion from inside



90-degrees zigzag-like motion



30-degrees zigzag-like motion

Selecting cleaning strategy

Example 1 : Vacuum cleaners

Question: How to choose a strategy

Selecting cleaning strategy

Example 1 : Vacuum cleaners

Question: How to choose a strategy

Possible Answer:

1. Let user to navigate the vacuum cleaner

2. Have several buttons on the vacuum cleaner to let the user to switch between the strategies (automatic VC)

3. Let vacuum cleaner choose the best strategy (AI solution)

Selecting cleaning strategy

Example 1 : Vacuum cleaners

Do you have more questions

Selecting cleaning strategy

Example 1 : Vacuum cleaners

Question: what are important things to consider if the user has to come up with best strategy

1. **What is the goal**
2. **What are the constraints**
3. **What is the surface**
4. **……**

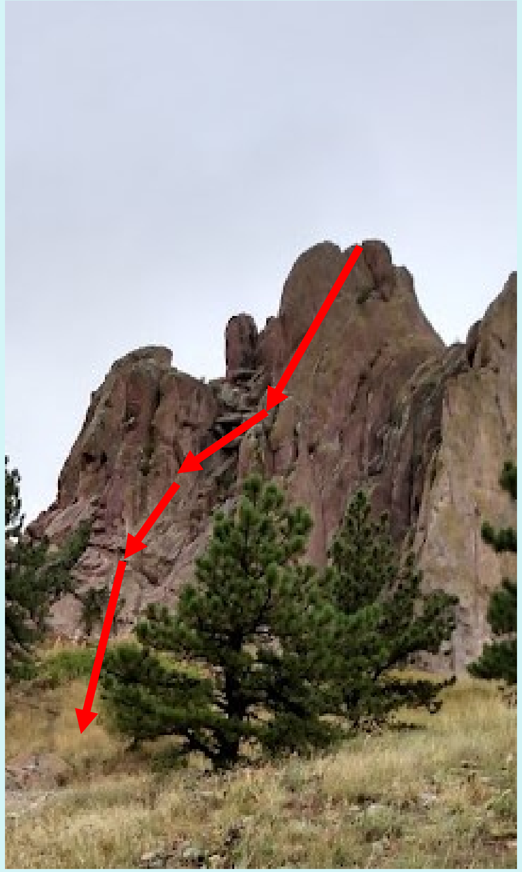
How easy is it to judge if selected strategy is the best? Or good enough? Or better than the other strategy? It would be nice to have a systematic approach to compare strategies!

Example 1 : Vacuum cleaners

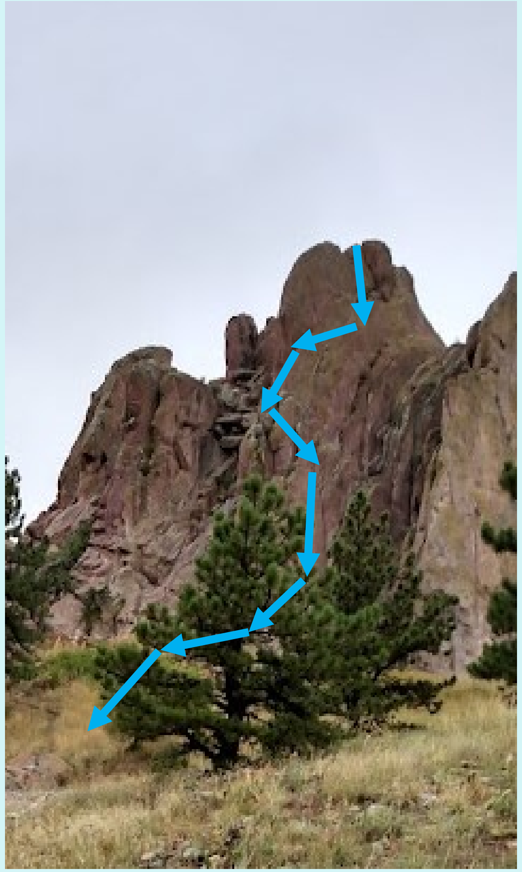
Questions about first example?

Selecting optimal path

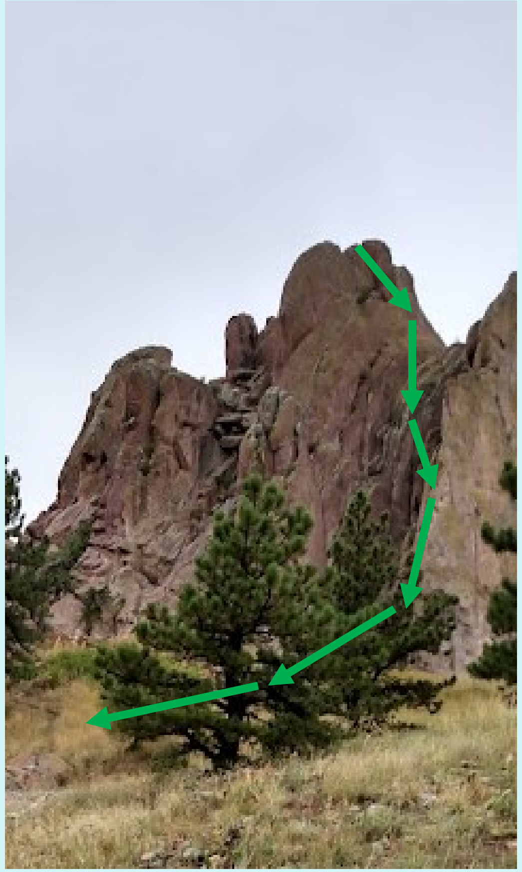
Example 2: find the best path down from the top of the mountain



Strategy 1



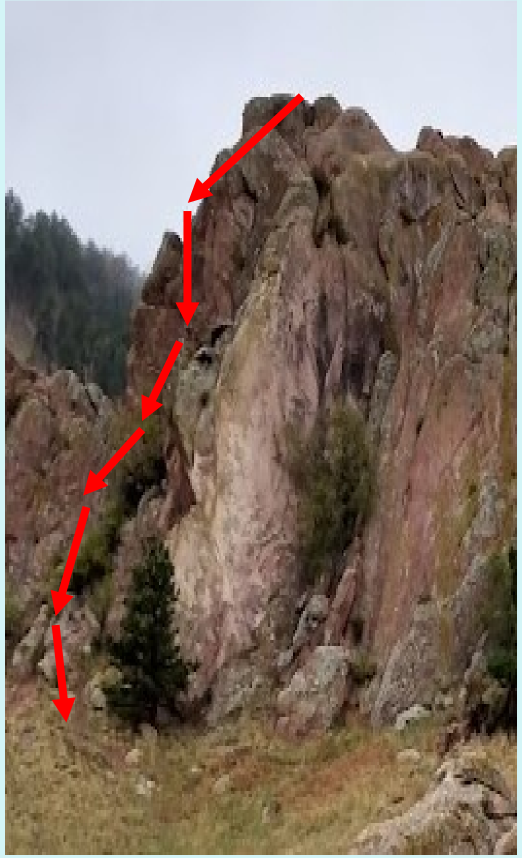
Strategy 2

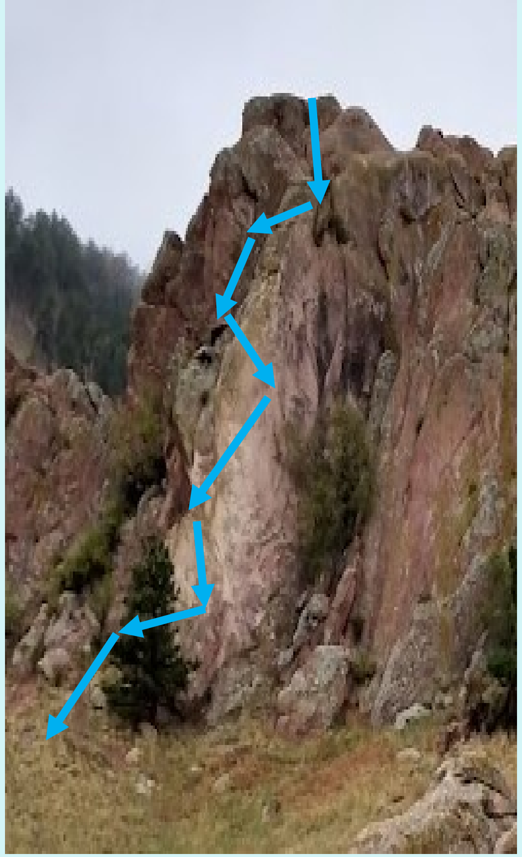


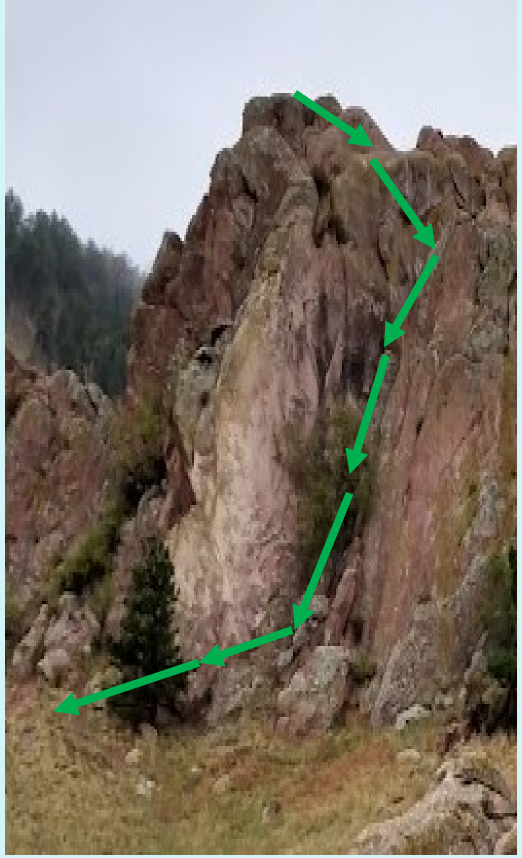
Strategy 3

Selecting optimal path

Example 2: find the best path down from the top of the mountain







Strategy 1 Strategy 2 Strategy 3

Selecting optimal path

Question: what are important things to consider if the user has to come up with best strategy to come down from the top of the mountain

1. **What is the goal**
2. **What are the constraints**
3. **What is the surface**
4. **……**

How easy is it to judge if selected strategy is the best? Or good enough? Or better than the other strategy? It would be nice to have a systematic approach to compare strategies!

Our software

why

1. It would be nice to have a systematic approach to compare strategies!
2. It would be nice to have a way to teach someone to apply and compare such strategies!

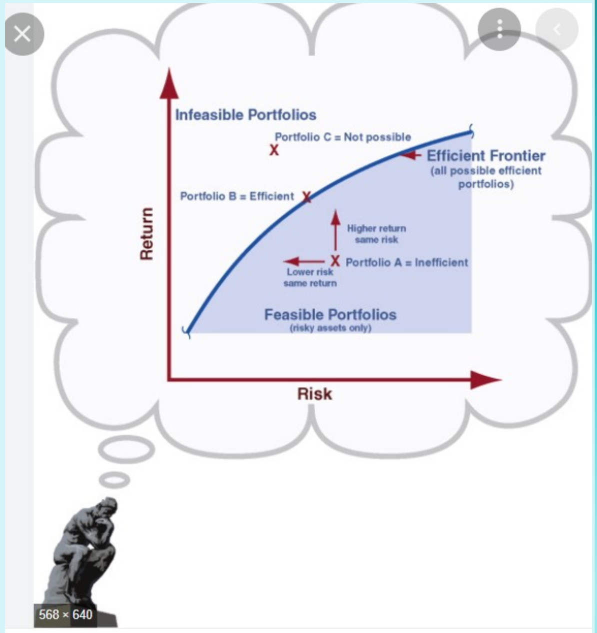
how

1. Our software is based on the systematic approach to construct global optimization methods (in other words, strategies)
2. Our software allows to construct your own strategy based on standard modules provided in the library

Real word example from finance

More complicated (100+ dimensions)

Portfolio Selection
(standard problem in investment)

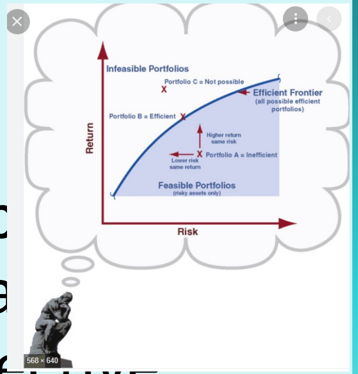


In portfolio selection one finds/compares *investment* strategies systematically!

Our software can do this, too!

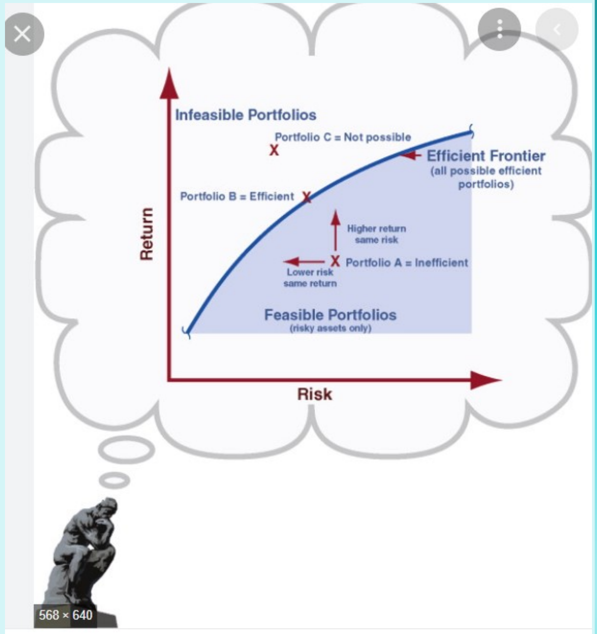
What is portfolio selection

- The portfolio selection problem is concerned with finding an op portfolio x of assets from a given set of n risky assets out of a specified asset universe such that the requirements of the respective investor are met. In general, investors seek to optimize their portfolio in regard of the trade-off between return and risk, such that the meta optimization problem can be formulated as



- minimize Risk(x)
- maximize Return(x)
- This bi-criteria optimization problem is commonly reduced to a single-criteria problem by just focusing on the risk and constraining the required mean, i.e. the investor sets a lower expected return target μ, minimize Risk(x) subject to Return(x) ≥ μ

What is portfolio selection



Measure of risk:

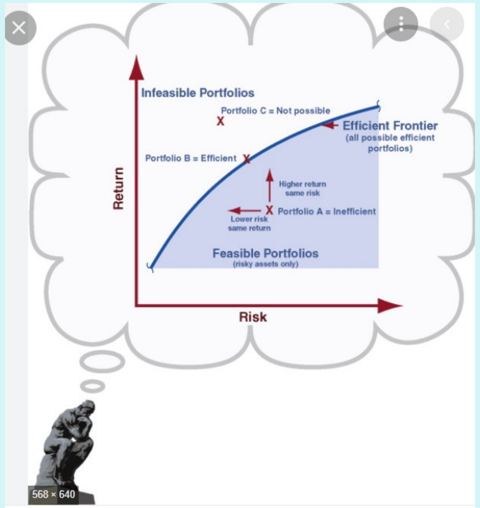
- L1: Mean Absolute Deviation (MAD)
- L2: Standard Deviation
- Linf: Maxmin
- Distribution based: VaR (value-at-Risk), CVaR (Expected shortfall), Omega
- Entropy based: Rho
- Risk Parity

Classical approach - Mean Variance Optimization

(Happy Marriage of L2 Risk and Assumption of Normality of Returns)

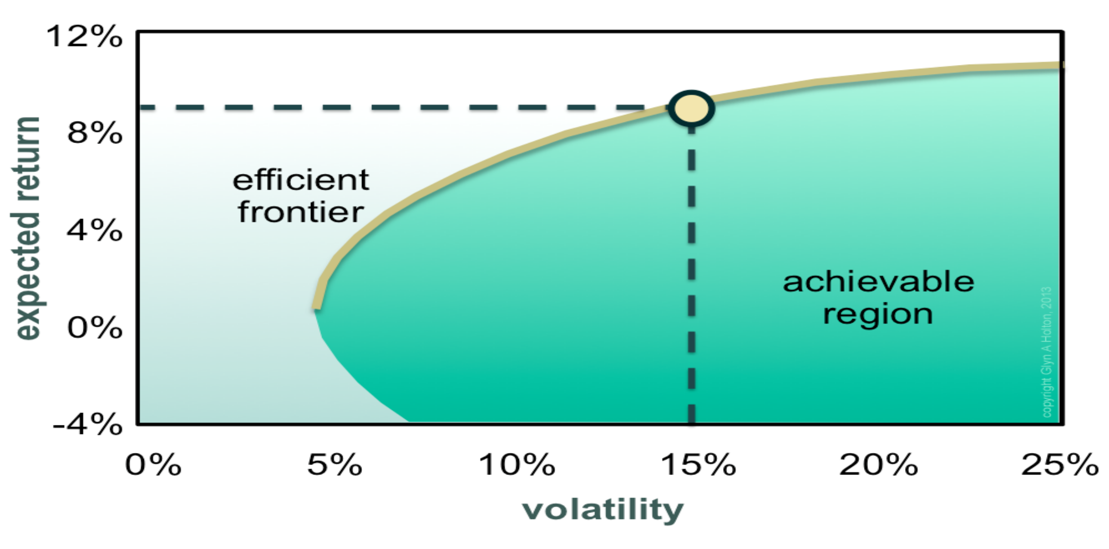
- Portfolio Volatility as Measure of Risk
- Returns of all market assets are assumed to be multivariate normal with the given vector of means and covariance matrix
- Solution is easy to find

What is portfolio selection

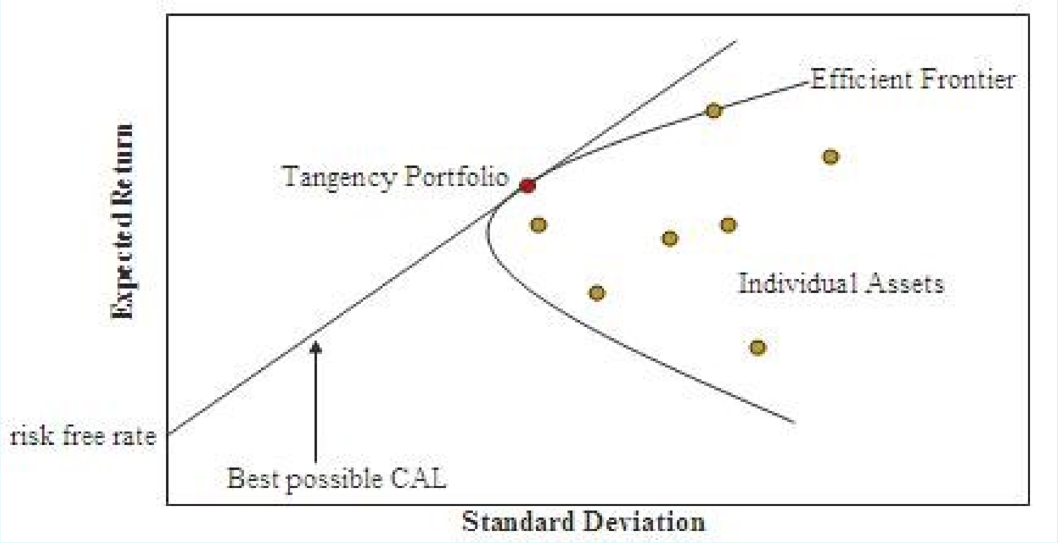


Classical approach - Mean Variance Optimization

Markowitz bullet



Tangency portfolio

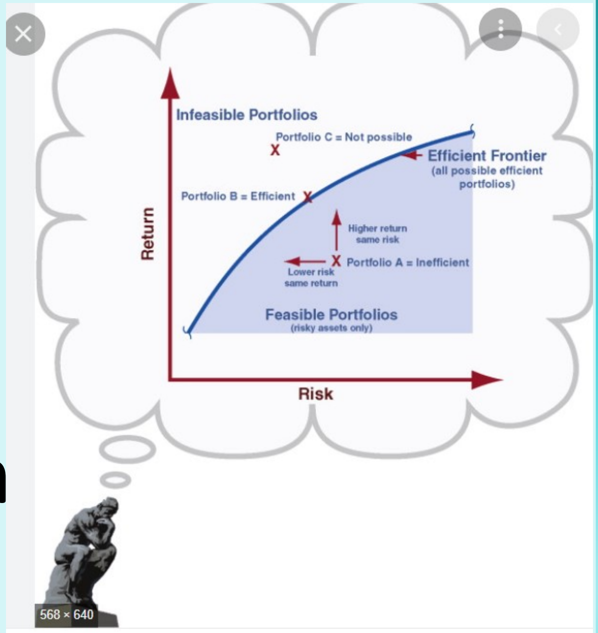


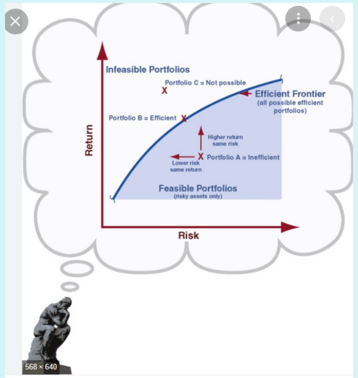
What is portfolio selection

Classical approach - Mean Variance Optimization

Solving quadratic optimization problem poses num

stability challenge!





What is portfolio selection

Ways to correct classical approach

Relax the assumptions of joint normality of returns (possibly take into account higher moments)

Use another risk measure, e.g. MAD, or CVAR

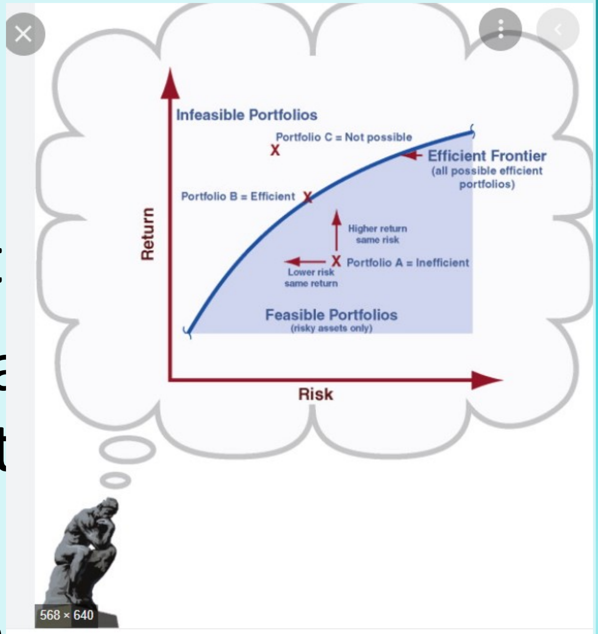
Switching to L1 doesn't help much, portfolios obtained using estimates from a limited set of scenarios are still performing poorly…

One important technique used for practical portfolio purposes are risk-parity portfolios, where the assets are weighted such that they equally contribute risk to the overall risk of the portfolio. **Within this approach the** focus is on allocation of risk, usually defined as volatility, rather than allocation of capital. The risk parity approach asserts that when asset allocations are adjusted (leveraged or deleveraged) to the same risk level, the risk parity portfolio can achieve a higher Sharpe ratio and can be more resistant to market downturns than the traditional portfolio.

Roughly speaking, the approach of building a risk parity portfolio is similar to creating a minimum-variance portfolio subject to the constraint that each asset (or asset class, such as bonds, stocks, real estate, etc.) contributes equally to the portfolio overall volatility.

What is portfolio selection

- It is sometimes difficult to disentangle the formulation optimization problem and the technicalities of its solut
- Approaches to portfolio optimization become more pra maybe we don’t need to seek the optimal solution, but a good one…
- As a result, there is a clear trend to use heuristic optimization methods, and we need a systematic way of generating and combining them…



Next slides - More about our software

Global optimization software library for research and education

- Created for students and researchers
- Has its roots in potential theory
- Optimization methods with desired properties are created based on basic modules, by varying parameters of generalized algorithm
- Allows:
  - to design optimization methods with guaranteed convergence but without the use of derivatives of objective function
  - to compare existing optimization methods
  - to understand principles of learning algorithms
  - to design custom variations or hybrids of known heuristic optimization methods.

Connecting Introductory example with general intent of our software MinPy

Optimization lies at the heart of machine learning and data science.
One of the most relevant problems in machine learning is automatic selection of optimization algorithm depending on the objective.
This is necessary in many applications such as robotics, simulating biological or chemical processes, trading strategies optimization, to name a few.

Connecting Introductory example
with general intent of our software MinPy

Optimization methods in this library work with all objectives including very onerous ones, such as black box functions and functions given by computer code, and the convergence of methods is guaranteed.

Connecting Introductory example
with general intent of our software MinPy

This library allows to create customized derivative free learning algorithms with desired properties by combining building blocks from this library or other Python libraries.

MinPy code Organization

Minimization – most important class in MinPy.
Its methods are building blocks for creating your own optimization strategies.

Attributes

- dim
- distribution
- initial_guess
- n_max_iterations
- n_points
- objective_function
- step_size
- tolerance

Methods

- Initialize()
- contract()
- reflect()
- shrink()
- update_m()
- update_c()
- stop()

MinPy code Organization

A new strategy should be created as follows:
Create a class that inherits from Minimization class.
Add a method that implements your strategy. It should rely on building blocks from Minimization class that can be used as is or be overwritten.

For example, below is the code that implements NM_minimization strategy

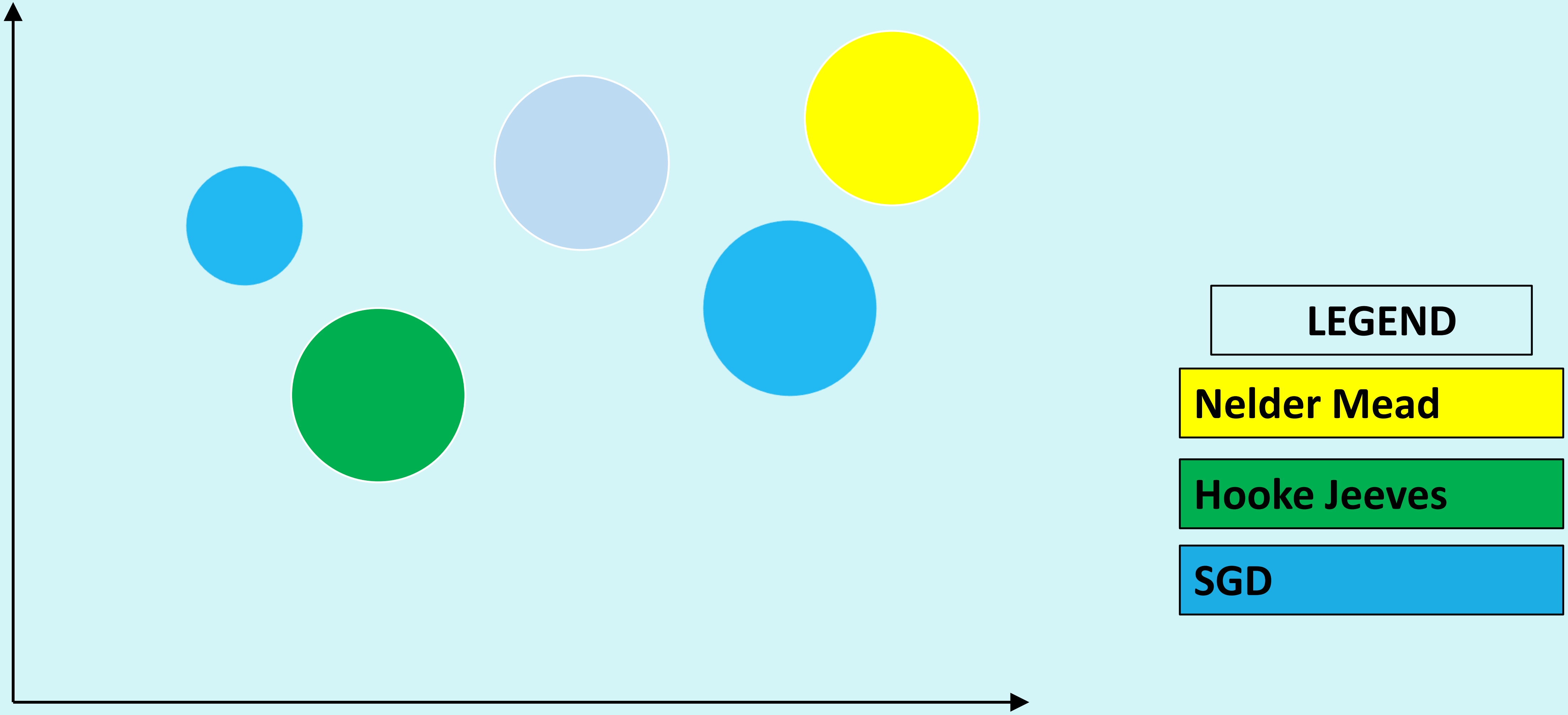
```python
class NM_Minimization(minpy.Minimization):


    def NM_stochastic(self):
        self.initialize()
        self.update_m()
        f_m_current = self.get_f(self.m)
        self.compute_fu()
        count = 0

        while not self.stop() and count<=self.maxIter:

            self.update_c()
            self.estimate_simplex()
            self.update_m()
            if self.get_f(self.m) >=f_m_current:
                self.adjust_step(f_m_current)
            count = count + 1
            f_m_current = self.get_f(self.m)
        return self.get_best()
```

Optimization methods classification



Optimization methods classification